UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/718,567	11/22/2000	Joseph M. Joy	205962	3107

23460 7590 04/05/2004

LEYDIG VOIT & MAYER, LTD
TWO PRUDENTIAL PLAZA, SUITE 4900
180 NORTH STETSON AVENUE
CHICAGO, IL 60601-6780

EXAMINER	
ALI, SYED J	

ART UNIT	PAPER NUMBER
2127	4

DATE MAILED: 04/05/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PR4

Office Action Summary	Application No.	Applicant(s)	
	09/718,567	JOY ET AL.	
	Examiner	Art Unit	
	Syed J Ali	2127	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 22 November 2000.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-68 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-68 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on 22 November 2000 is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-68 are pending in this application.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 9, 35-37, 56, 59-61, and 65-67 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

As per claims 9, 35-37, 56, and 60-61, it is not clearly understood whether they are independent or dependent claims. As is, computer-readable medium claims cannot be dependent on method claims.

As per claims 59-60, they are dependent on claim 61, which is a computer-readable medium claim, while claims 59-60 recite the "method of claim 61".

As per claim 65, it is not clearly understood whether it is an independent or dependent claim. As is, computer-readable medium claims cannot be dependent on system claims.

As per claims 66-67, the limitation "a third data structure" is indefinite. Specifically, there is no "second data structure" in the claims or in the respective parent claims. It is hereinafter interpreted that the limitation was meant to read "a third data field".

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 1-2, 7-15, 19-20, 25-29, 34-37, 42-45, 47-49, and 62-65 are rejected under 35 U.S.C. 102(e) as being anticipated by Howland et al. (USPN 6,018,741) (hereinafter Howland).

As per claim 1, Howland teaches the invention as claimed, including a method for capturing information about a structure of a process as it develops, the method comprising:

creating a root task object (col. 3 lines 44-53);

associating code for executing the process with the root task object (col. 7 lines 34-48);

executing the process by invoking the code associated with the root task object (col. 7 lines 34-48); and

whenever a parent task spawns a child task,

creating a task object for the child task (col. 3 lines 44-53);

associating the child task object with a task object of the parent (col. 3 lines 44-53);

associating code for executing the child task with the child task object (col. 7 lines 34-48); and

executing the child task by invoking the code associated with the child task object (col. 7 lines 34-48).

As per claim 2, Howland teaches the invention as claimed, including the method of claim 1 wherein associating the child task object with the parent task object includes setting a pointer in the child task object to point to the parent task object (col. 4 lines 51-58).

As per claim 7, Howland teaches the invention as claimed, including the method of claim 1 further comprising:

associating completion code with a task object (col. 11 lines 27-47); and

executing the completion code when a task associated with the task object completes (col. 11 lines 27-47).

As per claim 8, Howland teaches the invention as claimed, including the method of claim 7 further comprising:

deleting the task object when the completion code executes (col. 11 lines 27-47).

As per claim 9, Howland teaches the invention as claimed, including a computer-readable medium containing instructions for performing the method of claim 1 (col. 1 lines 18-50).

As per claim 10, Howland teaches the invention as claimed, including a computer-readable medium having stored thereon a first data structure, the first data structure comprising:

a first data field containing data representing code (col. 7 line 50 - col. 9 line 27); and

a second data field containing data representing an association of the first data structure with a second data structure (col. 10 line 58 - col. 11 line 26).

As per claim 11, Howland teaches the invention as claimed, including the first data structure of claim 10 wherein the first data field represents code by pointing to the code (col. 7 line 50 - col. 9 line 27).

As per claim 12, Howland teaches the invention as claimed, including the first data structure of claim 10 wherein the second data field represents the association of the first data structure with the second data structure by pointing to the second data structure (col. 10 line 58 - col. 11 line 26).

As per claim 13, Howland teaches the invention as claimed, including the first data structure of claim 10 wherein the second data structure is the parent of the first data structure (col. 10 line 58 - col. 11 line 26).

As per claim 14, Howland teaches the invention as claimed, including the first data structure of claim 10 wherein the code is code for executing a task (col. 7 line 50 - col. 9 line 27).

As per claim 15, Howland teaches the invention as claimed, including the first data structure of claim 14 further comprising:

a third data field containing data representing a state of the task (col. 7 line 50 - col. 9 line 27).

As per claim 19, Howland teaches the invention as claimed, including the first data structure of claim 14 further comprising:

a third data field containing data representing completion code (col. 11 lines 27-47).

As per claim 20, Howland teaches the invention as claimed, including the first data structure of claim 10 wherein the code is deletion code (col. 11 lines 27-47).

As per claim 25, Howland teaches the invention as claimed, including the first data structure of claim 10 further comprising:

a third data field containing data representing a location in source code of a task (col. 7 line 50 - col. 9 line 27).

As per claim 26, Howland teaches the invention as claimed, including the first data structure of claim 10 further comprising:

a third data field containing data representing a resource associated with the first data structure (col. 7 line 50 - col. 9 line 27).

As per claim 27, Howland teaches the invention as claimed, including the computer-readable medium of claim 10 further comprising:

the second data structure, wherein the second data structure comprises:

a third data field containing data representing code (col. 7 line 50 - col. 9 line 27); and

a fourth data field containing data representing an association of the second data structure with a third data structure (col. 10 line 58 - col. 11 line 26).

As per claim 28, Howland teaches the invention as claimed, including a method for capturing a structure of an association of resources as it develops, the method comprising:

creating a root resource object (col. 3 lines 44-53); and

whenever a parent resource spawns a child resource,

creating an object for the child resource (col. 3 lines 44-53); and

associating the child resource object with an object of the parent resource (col. 3 lines 44-53).

As per claim 29, Howland teaches the invention as claimed, including the method of claim 28 wherein associating the child resource object with the parent resource object includes setting a pointer in the child resource object to point to the parent resource object (col. 4 lines 51-58).

As per claim 34, Howland teaches the invention as claimed, including the method of claim 28 further comprising:

associating deletion code with a resource object (col. 11 lines 27-47); and

executing the deletion code when the resource is deleted (col. 11 lines 27-47).

As per claim 35, Howland teaches the invention as claimed, including a computer-readable medium containing instructions for performing the method of claim 28 (col. 1 lines 18-50).

As per claim 36, Howland teaches the invention as claimed, including a computer-readable medium having stored thereon a data structure comprising a resource object created by the method of claim 28 (col. 7 line 50 - col. 11 line 47).

As per claim 37, Howland teaches the invention as claimed, including a computer-readable medium having stored thereon a data structure comprising an association of resource objects created by the method of claim 28 (col. 7 line 50 - col. 11 line 47).

As per claim 42, Howland teaches the invention as claimed, including a computer-readable medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing members of a group (col. 7 line 50 - col. 9 line 27); and

a second data field containing data representing a function for manipulating members of the group (col. 7 line 50 - col. 11 line 47).

As per claim 43, Howland teaches the invention as claimed, including the data structure of claim 42 wherein group members are represented by pointers to the group members (col. 7 line 50 - col. 9 line 27).

As per claim 44, Howland teaches the invention as claimed, including the data structure of claim 42 wherein a group member is a software object (col. 7 line 50 - col. 9 line 27).

As per claim 45, Howland teaches the invention as claimed, including the data structure of claim 42 wherein the function in the second data field is in the set: add member to group, delete member from group, get next group member, apply a second function to group members, search for group member, create hash of key of group member, compare key with key of group member (col. 7 line 50 - col. 11 line 47).

As per claim 47, Howland teaches the invention as claimed, including the data structure of claim 42 further comprising:

a third data field containing data representing a second function for manipulating the group (col. 7 line 50 - col. 9 line 27).

As per claim 48, Howland teaches the invention as claimed, including the data structure of claim 47 wherein the second function is in the set: initialize group, de-initialize group (col. 7 line 50 - col. 9 line 27; col. 11 lines 27-47).

As per claim 49, Howland teaches the invention as claimed, including the data structure of claim 48 wherein the second function is de-initialize group and the second function does not de-initialize the group until the group contains no members and no functions for manipulating members of the group are active (col. 11 lines 27-47).

As per claim 62, Howland teaches the invention as claimed, including a system for capturing information about a structure of a process, the process comprising tasks, the system comprising:

a set of task objects, a task object corresponding to a task (col. 2 lines 27-32);

a data structure containing data representing associations among the task objects, the associations reflecting associations among the tasks (col. 2 lines 27-32); and

a set of functions callable by the tasks that build the data structure in response to changes in the associations among the tasks (col. 7 line 50 - col. 11 line 47).

As per claim 63, Howland teaches the invention as claimed, including the system of claim 62 wherein the data structure further contains data representing an association of a resource with a task (col. 7 line 50 - col. 9 line 27).

As per claim 64, Howland teaches the invention as claimed, including the system of claim 63 wherein a resource is in the set: task, section of memory, handle to system-supplied object, application-defined object, and lock (col. 7 line 50 - col. 9 line 27).

As per claim 65, Howland teaches the invention as claimed, including a computer-readable medium containing instructions for providing the system of claim 62 (col. 1 lines 18-50).

6. Claims 38-41 and 57-61 are rejected under 35 U.S.C. 102(e) as being anticipated by McDonald et al. (USPN 6,560,627) (hereinafter McDonald).

As per claim 38, McDonald teaches the invention as claimed, including a method for coordinating a task's interaction with an event, the method comprising:

associating a task object with the task (col. 2 lines 17-24);

when the task needs to wait for the event to occur,

suspending execution of the task and associating with the task object a resource object of a resource that causes the event, the association indicating that the task is waiting for the event to occur (col. 5 lines 30-37); and

when the event occurs,

searching the resource object for the association with the task object indicating that the task is waiting for the event to occur and, if found, resuming execution of the task (col. 5 lines 51-58).

As per claim 39, McDonald teaches the invention as claimed, including the method of claim 38 wherein the associating of the task object with the resource object comprises placing a reference to the task object in the resource object (col. 5 lines 30-37).

As per claim 40, McDonald teaches the invention as claimed, including the method of claim 39 wherein the associating of the task object with the resource object further comprises placing a reference to the resource object in the task object (col. 5 lines 18-29).

As per claim 41, McDonald teaches the invention as claimed, including the method of claim 38 further comprising:

when a task causes a second event,

searching the task object for an association with a second task object indicating that a second task is waiting for the second event and, if found, resuming execution of the second task (col. 5 lines 51-58).

As per claim 57, McDonald teaches the invention as claimed, including a method for coordinating a task's access to a resource, the method comprising:

creating a lock software object (col. 2 lines 36-46);

associating the lock software object with the resource (col. 2 lines 47-54);

when the task requests access to the resource, if the lock software object is not currently associated with another task, associating the lock software object with the task and granting the task access to the resource (col. 5 lines 18-29); and

when the task indicates that it no longer requires access to the resource, disassociating the lock software object from the task (col. 5 lines 18-29).

As per claim 58, McDonald teaches the invention as claimed, including the method of claim 57 further comprising:

associating a lock level with the lock software object (Claim 1);

associating a lock order with the lock software object (col. 4 lines 9-16); and

when the task requests access to the resource,

examining lock levels of other lock software objects associated with the task (col. 5 lines 38-50), and

denying the request if granting the request would violate the lock order, given the lock levels of the other lock software objects associated with the task (col. 5 lines 38-50).

As per claim 59, McDonald teaches the invention as claimed, including the method of claim 61 wherein functions for associating the lock software object with the task and disassociating the lock software object from the task are supplied by a creator of the task (col. 4 lines 1-8).

As per claim 60, McDonald teaches the invention as claimed, including a computer-readable medium containing instructions for performing the method of claim 61 (col. 1 lines 9-11).

As per claim 61, McDonald teaches the invention as claimed, including a computer-readable medium having stored thereon a data structure comprising a lock software object created by the method of claim 57 (col. 1 lines 9-11; col. 2 lines 36-46).

7. Claims 50-52 and 55-56 are rejected under 35 U.S.C. 102(b) as being anticipated by Katzenberger (USPN 5,970,496).

As per claim 50, Katzenberger teaches the invention as claimed, including a method for controlling the existence of a first software object, the method comprising:

initializing a reference counter associated with the first software object when the first software object is created (col. 15 lines 41-46);

incrementing the reference counter when a second software object that is a child of the first software object is created (col. 16 lines 15-25);

decrementing the reference counter when the child software object is deleted (col. 16 lines 43-53);

when a request is made to delete the first software object, denying the request unless the reference counter equals a predetermined value (Claim 11); and

when the reference counter equals the predetermined value, deleting the first software object (Claim 11).

As per claim 51, Katzenberger teaches the invention as claimed, including the method of claim 50 wherein the first software object is associated with a resource in a computing system (col. 2 line 66 - col. 3 line 9).

As per claim 52, Katzenberger teaches the invention as claimed, including the method of claim 51 wherein the resource is in the set: task, section of memory, handle to system-supplied object, application-defined object, and lock (col. 2 line 66 - col. 3 line 9).

As per claim 55, Katzenberger teaches the invention as claimed, including the method of claim 50 further comprising:

incrementing the reference counter on an indication that a need has arisen for the first software object (col. 16 lines 15-25); and

decrementing the reference counter on an indication that the need no longer exists (col. 16 lines 43-53).

As per claim 56, Katzenberger teaches the invention as claimed, including a computer-readable medium containing instructions for performing the method of claim 50 (col. 1 lines 6-11).

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 3, 16-18, 21-22, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Howland.

As per claims 3 and 30, "Official Notice" is taken that although Howland does not specifically teach associating the child task/resource object with the parent task/resource object by setting a pointer in the parent task/resource object to point to the child task/resource object, such would have been obvious to one of ordinary skill in the art. Specifically, Howland teaches exclusively of setting pointers such that the child points to the parent, and maintaining relationships in this manner. However, bi-directional pointers are well known and expected in the art, especially within object oriented data structures such as trees. Bi-directional pointers are relatively simple to implement, and allow more flexibility in algorithms for manipulation of trees.

As per claim 16, Howland teaches the invention as claimed, including the first data structure of claim 15 wherein the first data structure further comprises:

a fourth data field containing data representing an association of the first data structure with an object on whose deletion the task is pending (col. 11 lines 27-47).

"Official Notice" is taken that although Howland does not specify the state of the task is "pending on object deletion", the functionality achieved by the cited section of Howland provides the implementation to arrive at the claimed result of "pending on object deletion".

As per claim 17, Howland teaches the invention as claimed, including the first data structure of claim 15 wherein the first data structure further comprises:

a fourth data field containing data representing a second task on whose completion the task is pending (col. 7 line 50 - col. 9 line 27).

"Official Notice" is taken that although Howland does not specify the state of the task is "pending on task completion", the functionality achieved by the cited section of Howland provides the implementation to arrive at the claimed result of "pending on task completion".

As per claim 18, Howland teaches the invention as claimed, including the first data structure of claim 15 wherein the first data structure further comprises:

a fourth data field containing data representing a group on whose de-initialization the task is pending (col. 11 lines 27-47).

"Official Notice" is taken that although Howland does not specify the state of the task is "pending on group de-initialization", the functionality achieved by the cited section of Howland provides the implementation to arrive at the claimed result of "pending on group de-initialization".

As per claims 21-22, "Official Notice" is taken that although Howland does not specifically teach a third data field containing data representing an association of the first data structure with a third data structure and that the third data structure is a child of the first data structure, such would have been obvious to one of ordinary skill in the art. Specifically, Howland teaches exclusively of setting pointers such that the child points to the parent, and maintaining relationships in this manner. However, bi-directional pointers are well known and expected in the art, especially within object oriented data structures such as trees. Bi-directional pointers are relatively simple to implement, and allow more flexibility in algorithms for manipulation of trees.

10. Claims 4-6, 23-24, 31-33, and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Howland in view of Katzenberger.

As per claim 4, Katzenberger teaches the invention as claimed, including the method of claim 1 wherein associating the child task object with the parent task object includes incrementing a reference counter in the parent task object (Claim 10).

It would have been obvious to one of ordinary skill in the art to combine Howland and Katzenberger since utilizing a processor to check all elements of an array (or tree) to determine if it is a leaf node or if children exist for a node can be computationally intensive, and this workload can be alleviated through use of counters, such that a counter value of zero would indicate that a node is a leaf node.

As per claim 5, Katzenberger teaches the invention as claimed, including the method of claim 4 further comprising decrementing the reference counter in the parent task object when the child task object is deleted (Claim 11).

As per claim 6, Katzenberger teaches the invention as claimed, including the method of claim 4 further comprising deleting the parent task object when the reference counter reaches a predetermined value (Claim 11).

As per claim 23, Katzenberger teaches the invention as claimed, including the first data structure of claim 10 further comprising:

a third data field containing data representing a reference counter (Claims 10-11).

As per claim 24, Katzenberger teaches the invention as claimed, including the first data structure of claim 23 wherein the reference counter counts a number of child data structures that are associated with the first data structure (Claims 10-11).

As per claim 31, Katzenberger teaches the invention as claimed, including the method of claim 28 wherein associating the child resource object with the parent resource object includes incrementing a reference counter in the parent resource object (Claim 10).

As per claim 32, Katzenberger teaches the invention as claimed, including the method of claim 31 further comprising decrementing the reference counter in the parent resource object when the child resource object is deleted (Claim 11).

As per claim 33, Katzenberger teaches the invention as claimed, including the method of claim 31 further comprising deleting the parent resource object when the reference counter reaches a predetermined value (Claim 11).

As per claim 46, Katzenberger teaches the invention as claimed, including the data structure of claim 45 wherein a reference counter is associated with a group member, the function is delete member from group, and the function does not delete the member from the group if the member's reference counter does not equal a predetermined value (Claims 10-11).

11. Claims 53-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Katzenberger in view of Howland.

As per claim 53, Howland teaches the invention as claimed, including the method of claim 50 wherein the first software object is created and deleted by functions supplied by a creator of the first software object (col. 7 line 50 - col. 9 line 27).

It would have been obvious to one of ordinary skill in the art to combine Katzenberger and Howland since utilizing a processor to check all elements of an array (or tree) to determine if it is a leaf node or if children exist for a node can be computationally intensive, and this

workload can be alleviated through use of counters, such that a counter value of zero would indicate that a node is a leaf node.

As per claim 54, Katzenberger teaches the invention as claimed, including the method of claim 50 further comprising:

incrementing the reference counter when a cross reference is made between the first software object and another software object (col. 16 lines 15-25); and

decrementing the reference counter when a cross reference made between the first software object and another software object is deleted (col. 16 lines 43-53).

12. Claims 66 and 68 rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald in view of Katzenberger.

As per claim 66, McDonald teaches the invention as claimed, including a computer-readable medium having stored thereon a data structure, the data structure comprising:

a first data field containing data representing a count of locks held in the context of the call tree (col. 6 line 50 - col. 7 line 20).

Katzenberger teaches the invention as claimed, including a first data field containing data representing a count of temporary references existing in a context of a call tree (col. 15 lines 41-46).

It would have been obvious to one of ordinary skill in the art to combine McDonald and Katzenberger since utilizing a processor to check all elements of an array (or tree) to determine if

it is a leaf node or if children exist for a node can be computationally intensive, and this workload can be alleviated through use of counters, such that a counter value of zero would indicate that a node is a leaf node. The knowledge of whether or not a node is a leaf node in reference to locking mechanisms is useful for preventing deadlocks, such that multiple tasks do not acquire a shared resource by way of inheritance.

As per claim 68, McDonald teaches the invention as claimed, including the data structure of claim 66 further comprising:

a third data field containing data representing a group of locks held in the context of the call tree (col. 6 line 50 - col. 7 line 20).

13. Claim 67 is rejected under 35 U.S.C. 103(a) as being unpatentable over McDonald in view of Katzenberger in view of Howland.

As per claim 67, Howland teaches the invention as claimed, including the data structure of claim 66 further comprising:

a third data field containing data representing an assertion failure function (col. 7 line 50 - col. 11 line 47).

It would have been obvious to one of ordinary skill in the art to combine McDonald, Katzenberger, Howland for reasons discussed above in reference to claim 66, and also since handling exceptions or other failures in the manner taught by Howland decreases the likelihood of deadlock situations or misallocation of resources.

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

USPN 5,463,389 to Klayman teaches using counters to indicate the number of child nodes in a hierarchical array.

USPN 6,678,716 to Pronsati, Jr. et al. teaches a tree arrangement of processes that may be either software or non-software processes.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Syed J Ali whose telephone number is (703) 305-8106. The examiner can normally be reached on Mon-Fri 8-5:30, 2nd Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai T An can be reached on (703) 305-9678. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Syed Ali
March 31, 2004

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100